

Pontificia Universidad Católica de Chile  
Escuela de Ingeniería  
Departamento de Ciencia de la Computación

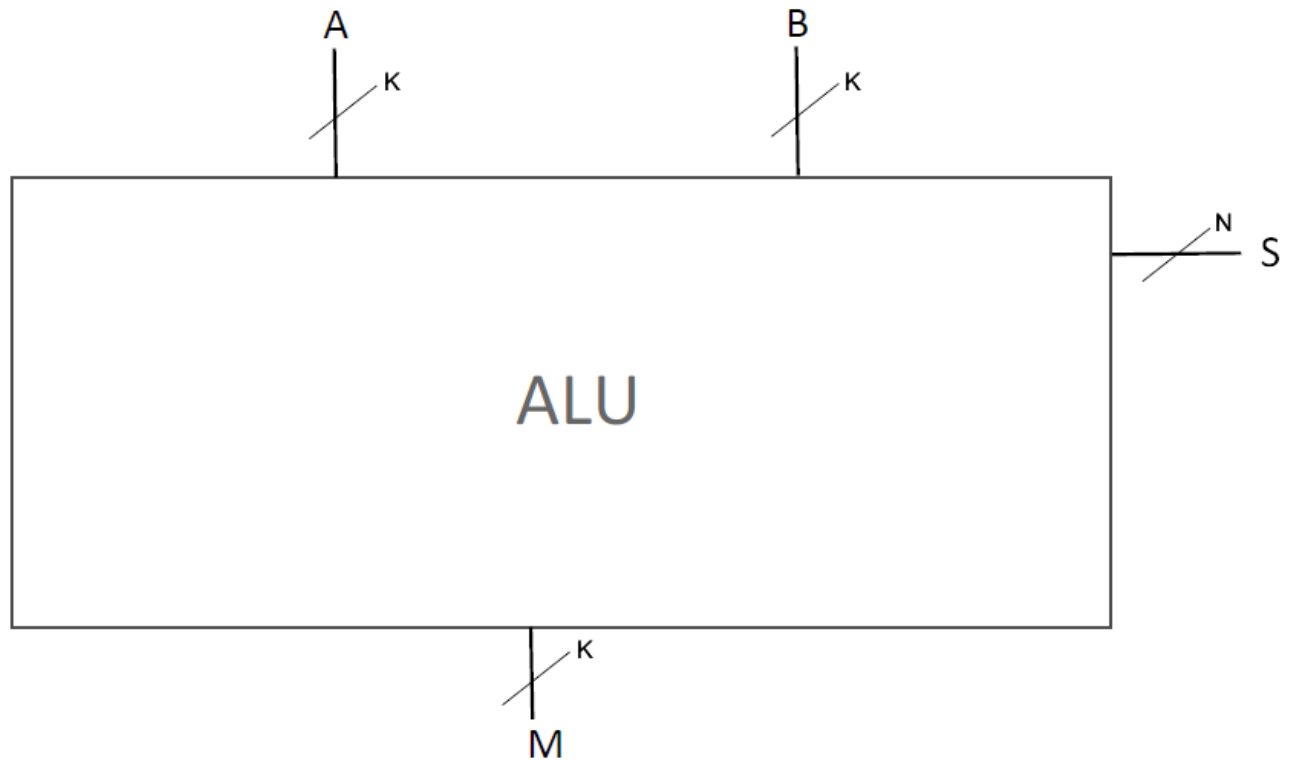


# IIC2343 – Arquitectura de Computadores

Almacenamiento de Datos

**Profesor:** Hans Löbel

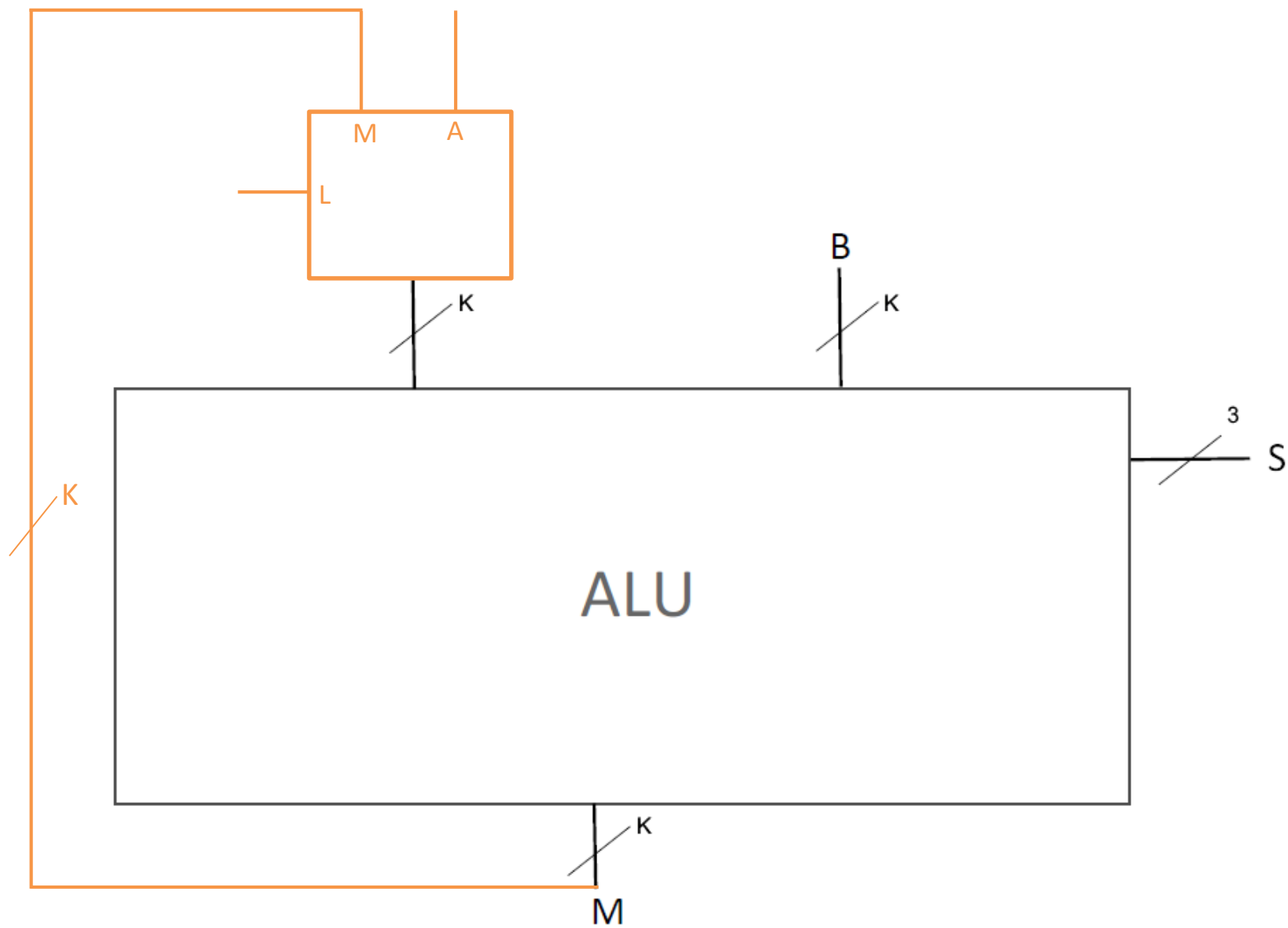
Esto es lo que tenemos hasta ahora



# Extendiendo las capacidades del **computador**

- Hasta el momento, nuestro **computador** no es más que una simple calculadora de nivel básico.
- ¿Es posible usar esta **calculadora**, tal como está ahora, para almacenar resultados previos?
- Una solución es conectar la **salida** de la ALU con la **entrada** A.

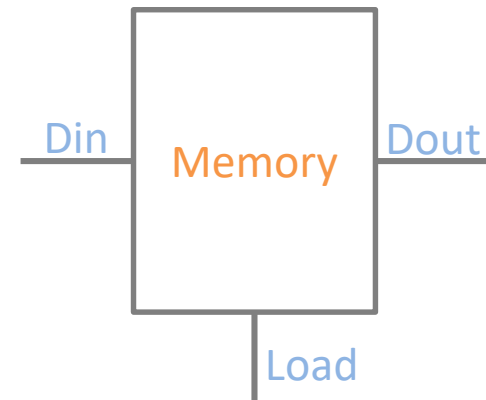




# Necesitamos una nueva pieza de hardware

Esta nueva pieza debe:

- Detener el resultado anterior (**almacenar datos**).
- Cambiar su estado (**datos almacenados**) sólo en un instante determinado.

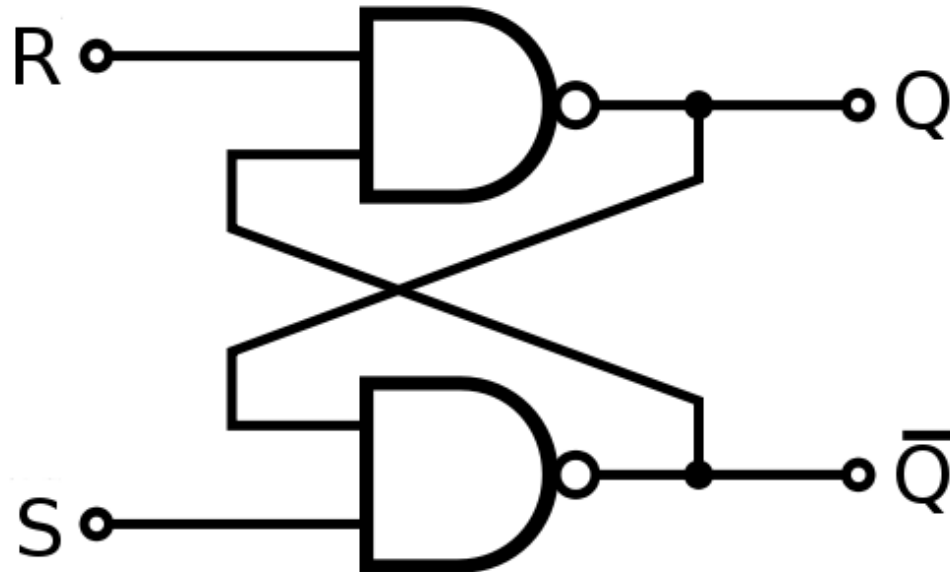


Latches y Flip-Flops son las piezas que necesitamos

- Fabricados con las mismas compuertas que conocemos.
- Distintos tipos dependiendo de la aplicación, ej: RS, D.
- Distintas combinaciones de estos permiten construir distintos tipos de memoria.

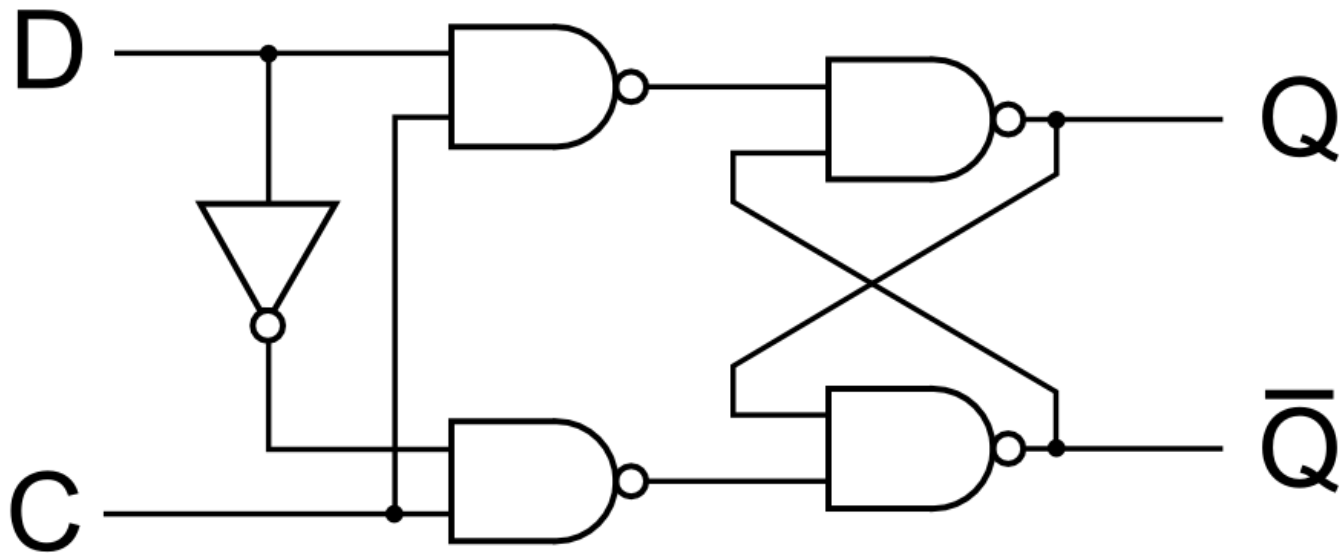
## Latch RS

| R | S | $Q^{t+1}$ |
|---|---|-----------|
| 0 | 0 | -         |
| 0 | 1 | 1         |
| 1 | 0 | 0         |
| 1 | 1 | $Q^t$     |

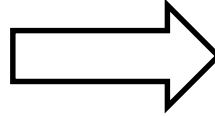
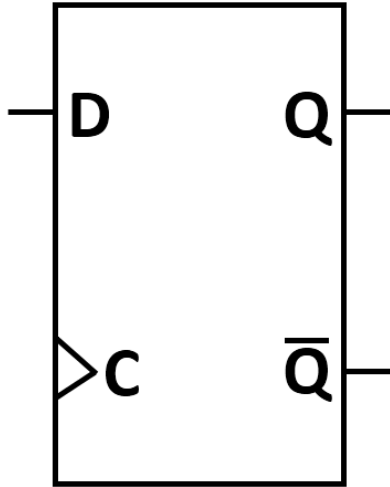


## Latch D

| C | D   | $Q^{t+1}$ |
|---|-----|-----------|
| 0 | 0/1 | $Q^t$     |
| 1 | 0   | 0         |
| 1 | 1   | 1         |





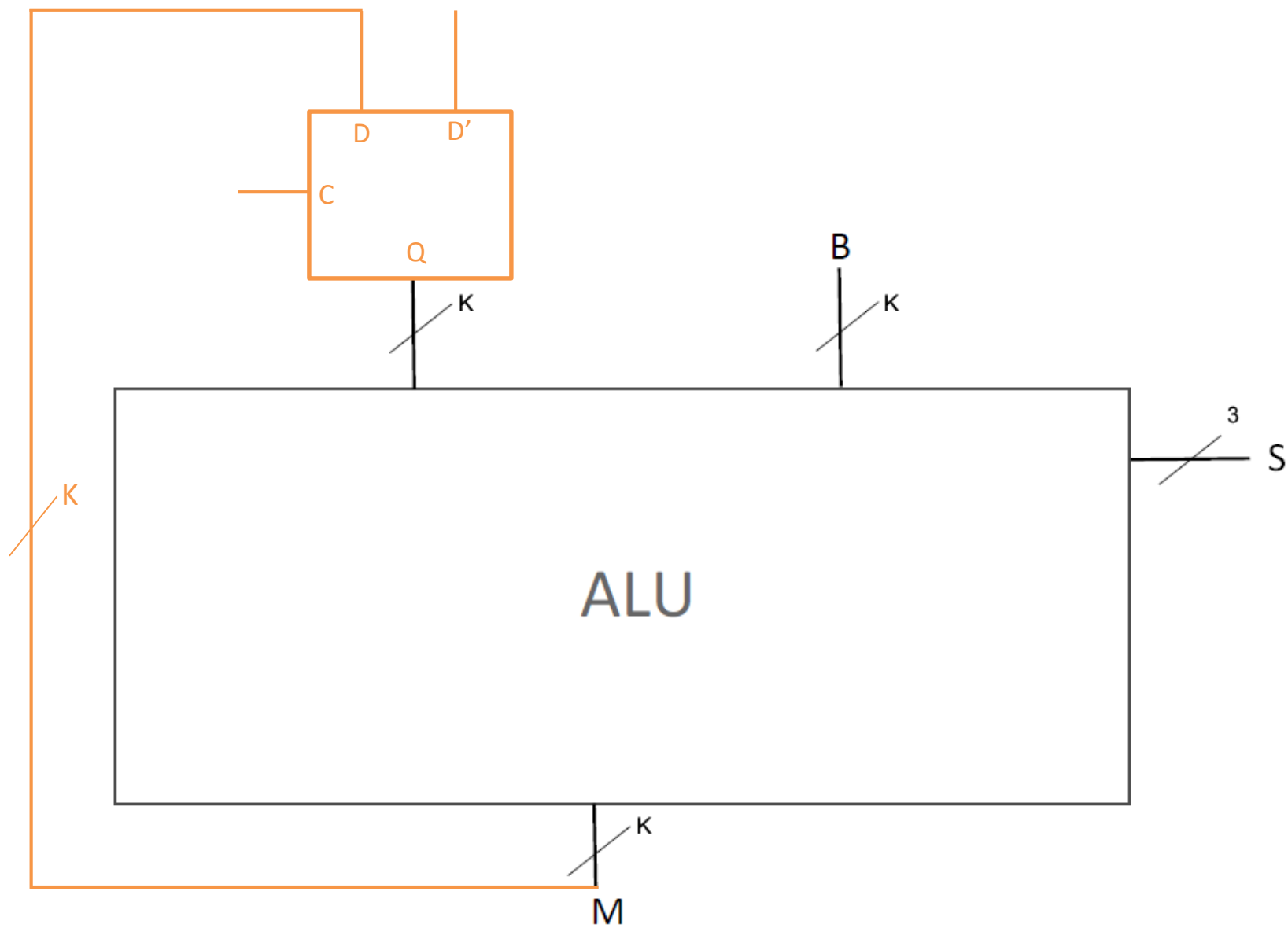


| C | D   | $Q^{t+1}$ |
|---|-----|-----------|
| 0 | 0/1 | $Q^t$     |
| 1 | 0   | 0         |
| 1 | 1   | 1         |

D: dato

C: control

Q: estado



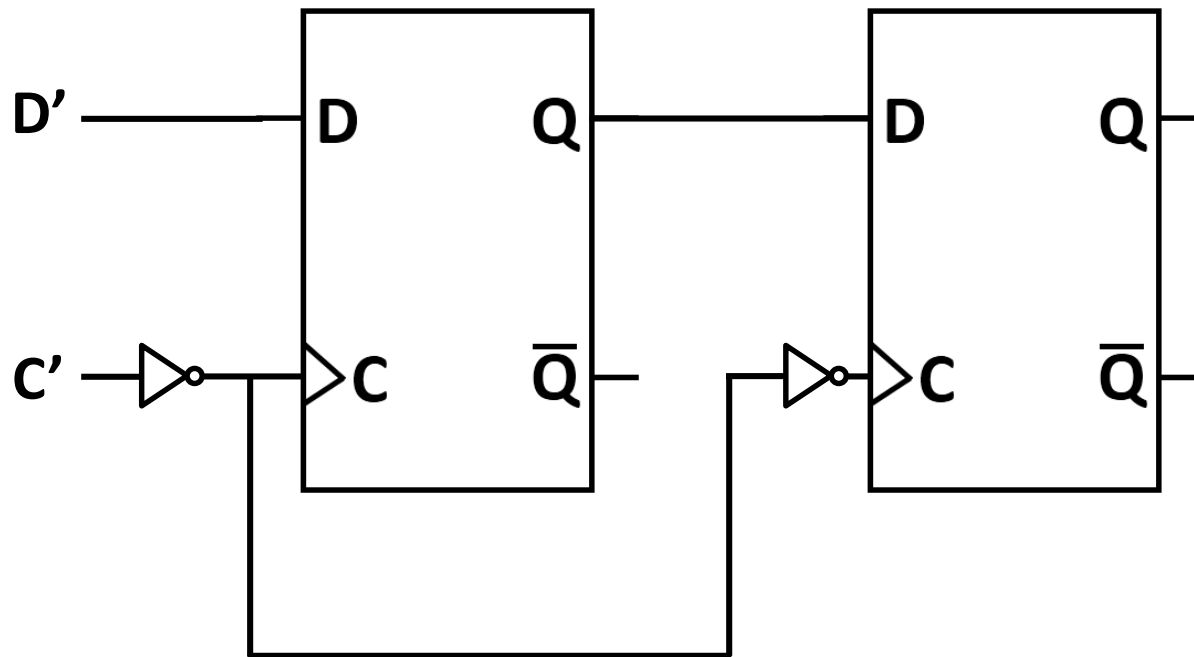
## El latch no es suficiente

- Los latches se activan en un estado (1 ó 0), lo que no soluciona del todo nuestros problemas.
- Necesitamos una pieza similar, pero que se active sólo en un instante dado.

¿Cuál podría ser ese instante?

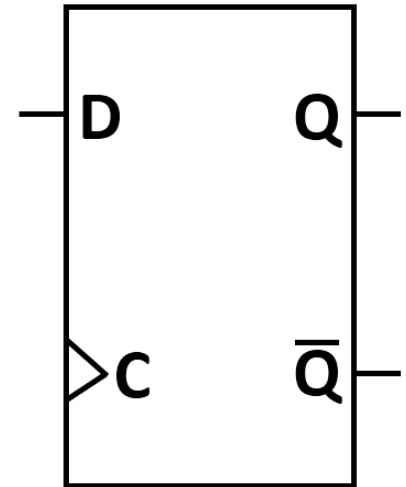
## Flip-Flop D

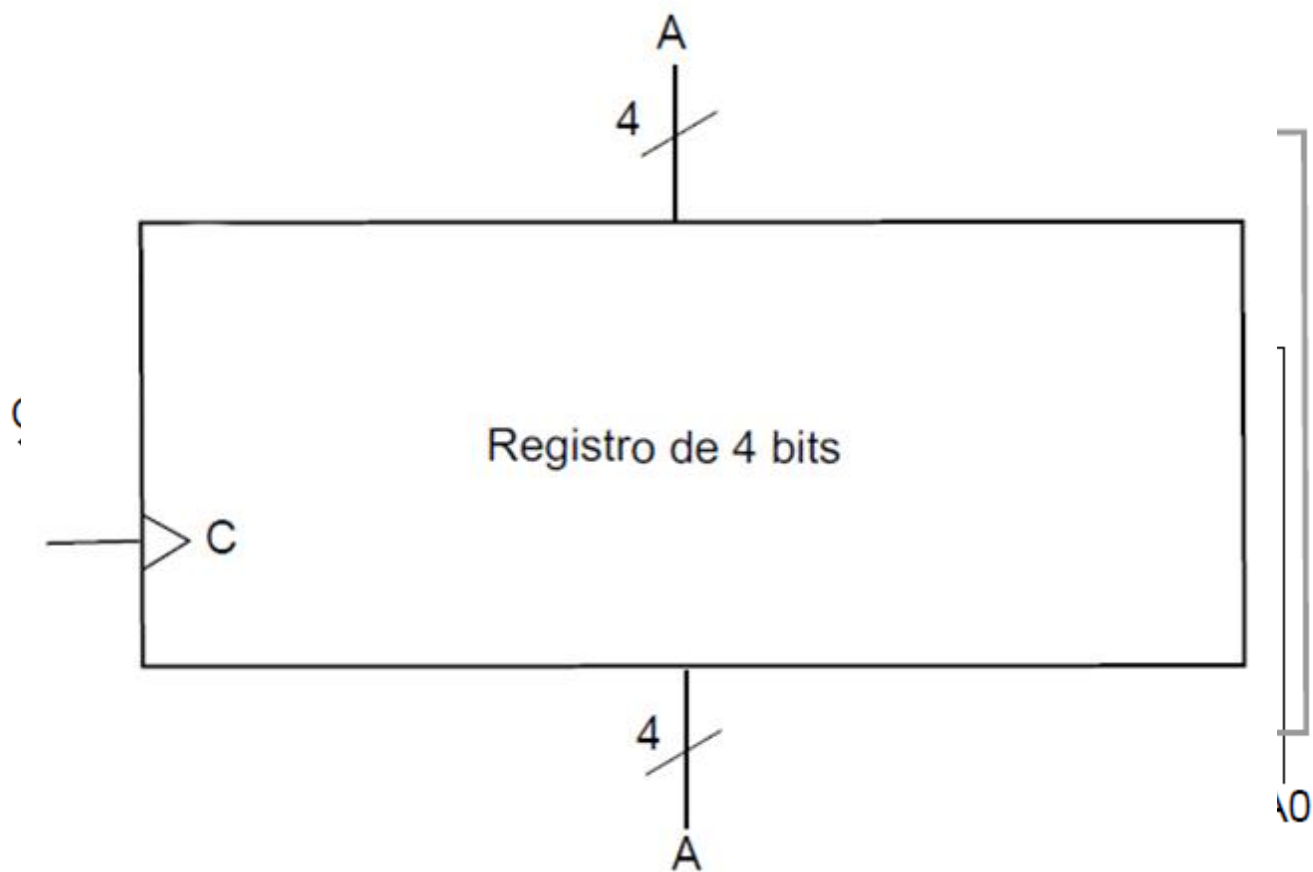
| C                 | D   | $Q^{t+1}$ |
|-------------------|-----|-----------|
| 0/1/ $\downarrow$ | 0/1 | $Q^t$     |
| $\uparrow$        | 0   | 0         |
| $\uparrow$        | 1   | 1         |

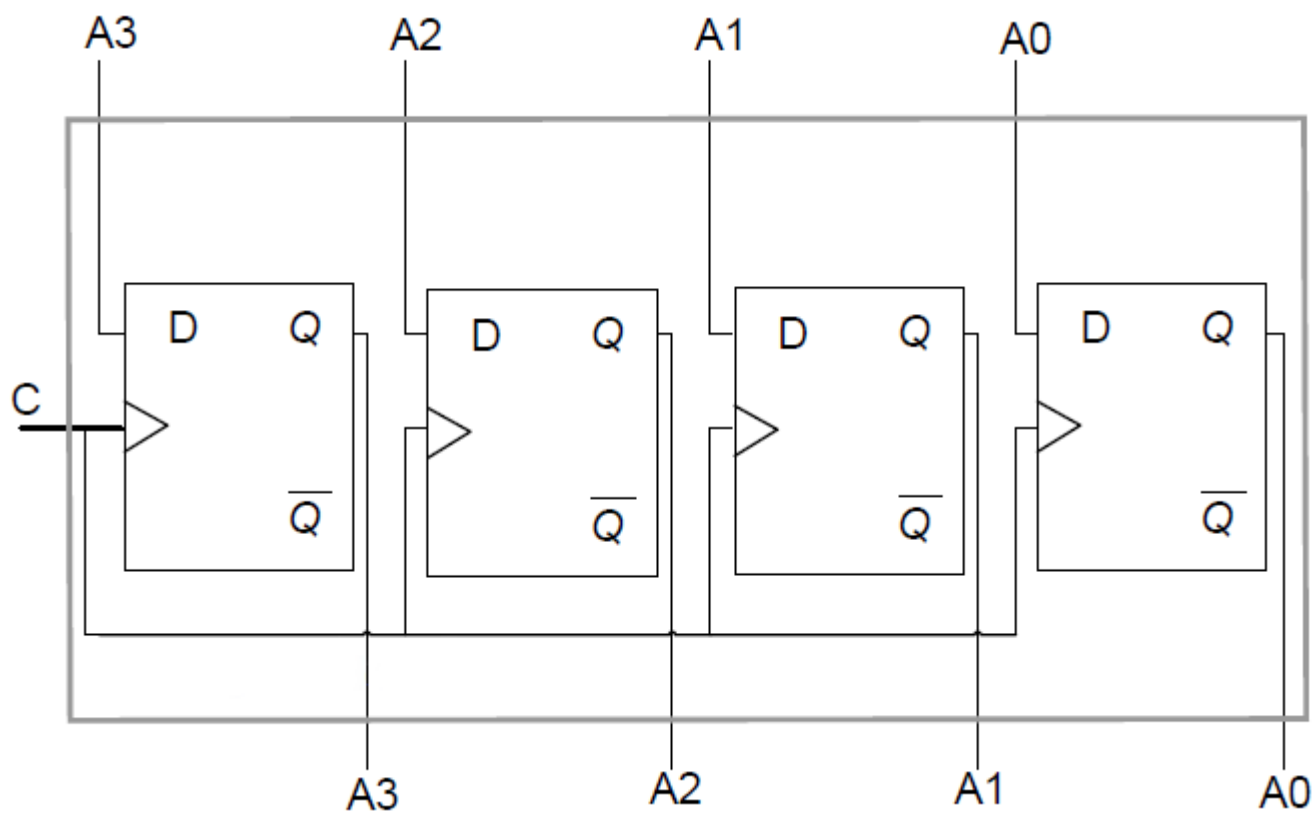


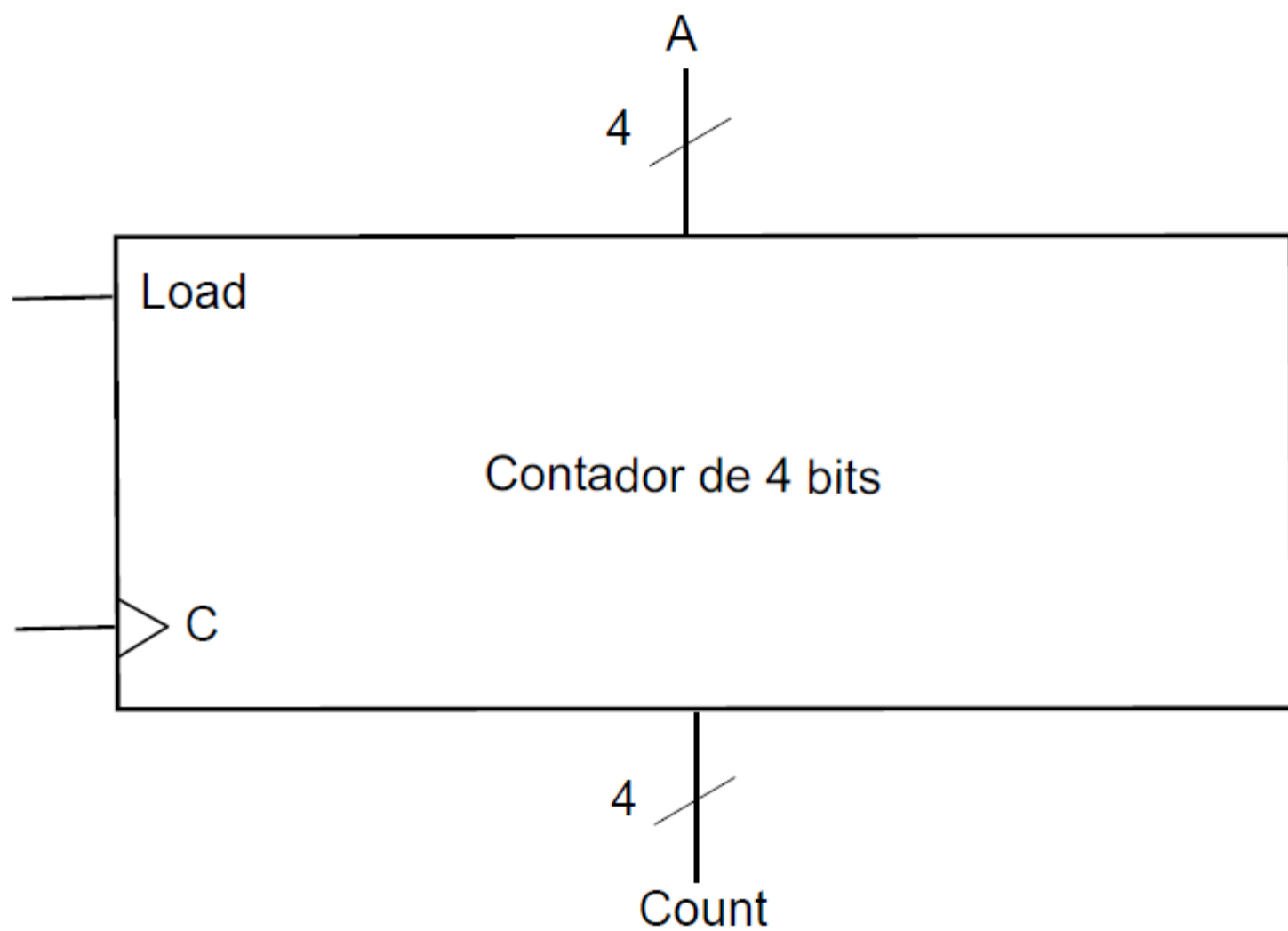
## Flip-Flops son la solución

- Los **flip-flops** se activan sólo en una transición o flanco (subida o bajada).
- De ahora en adelante, salvo que se diga explícitamente lo contrario, sólo hablaremos de **flip-flops** y estos se activarán por defecto **flanco de subida**.

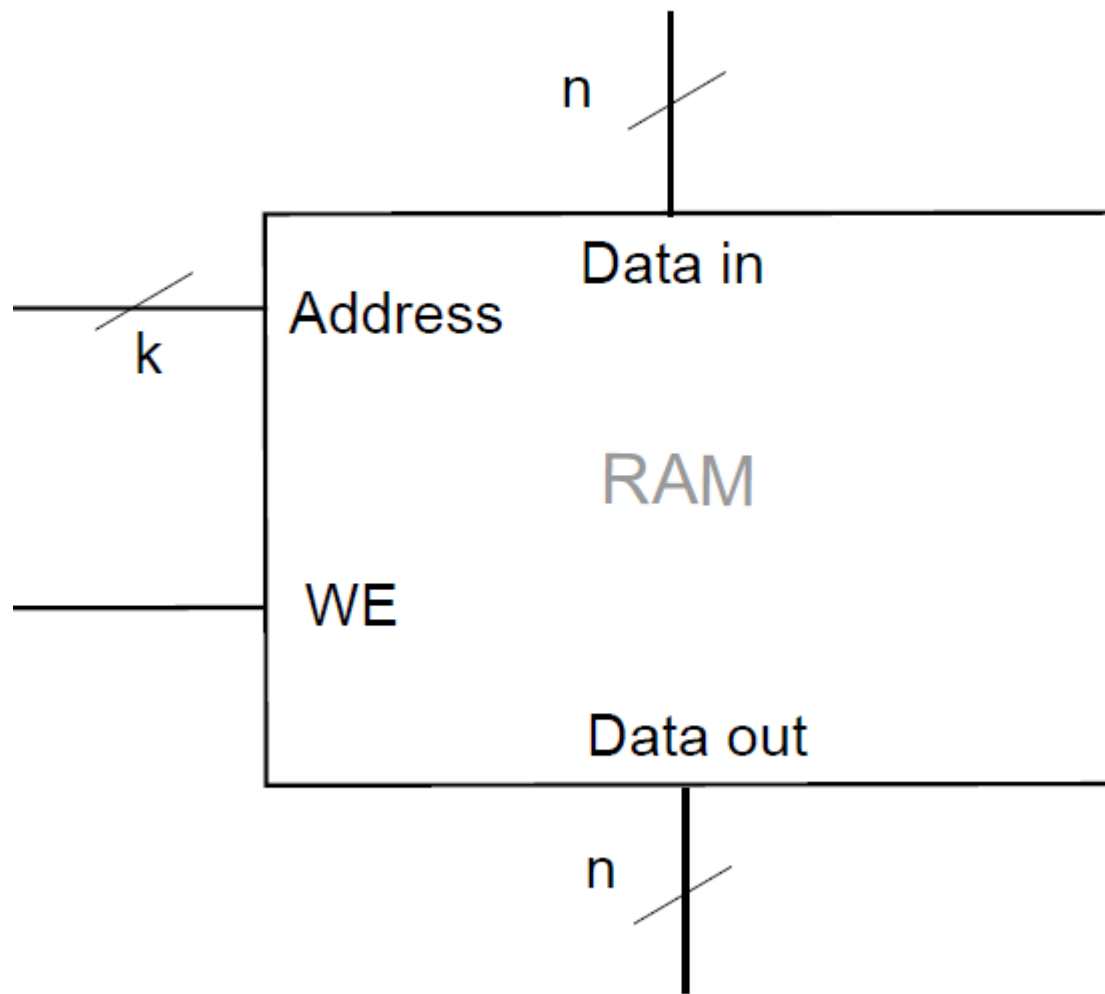


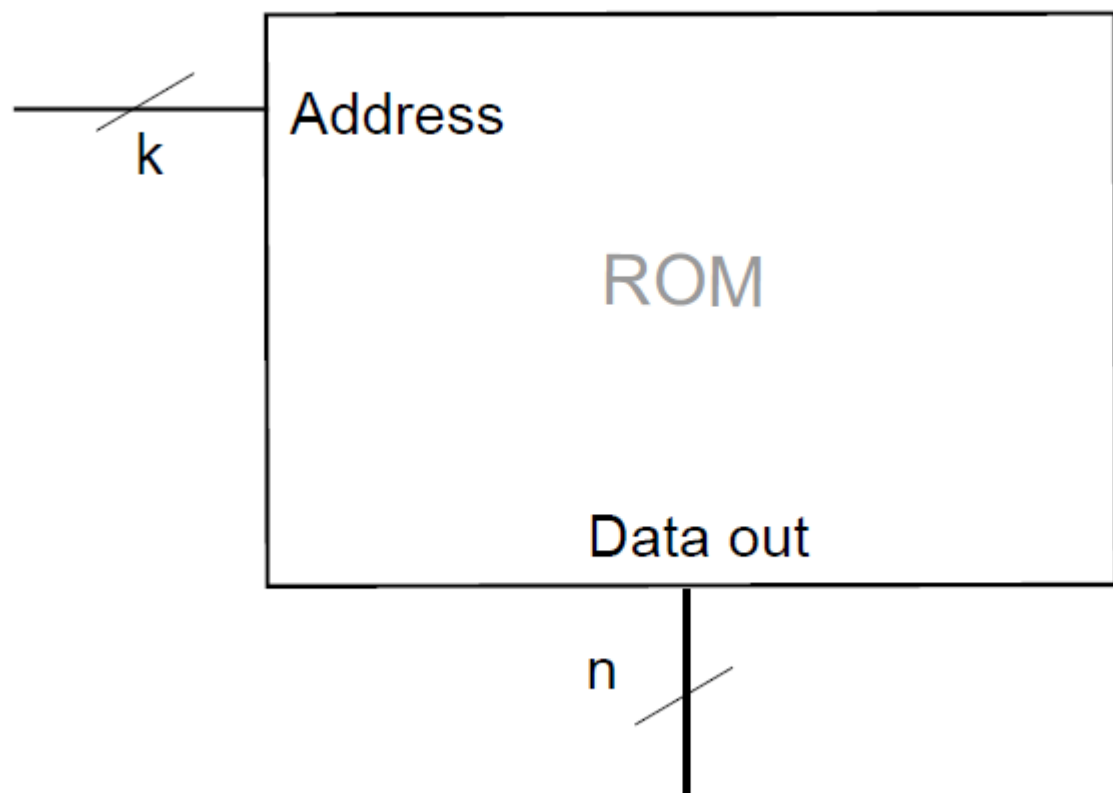




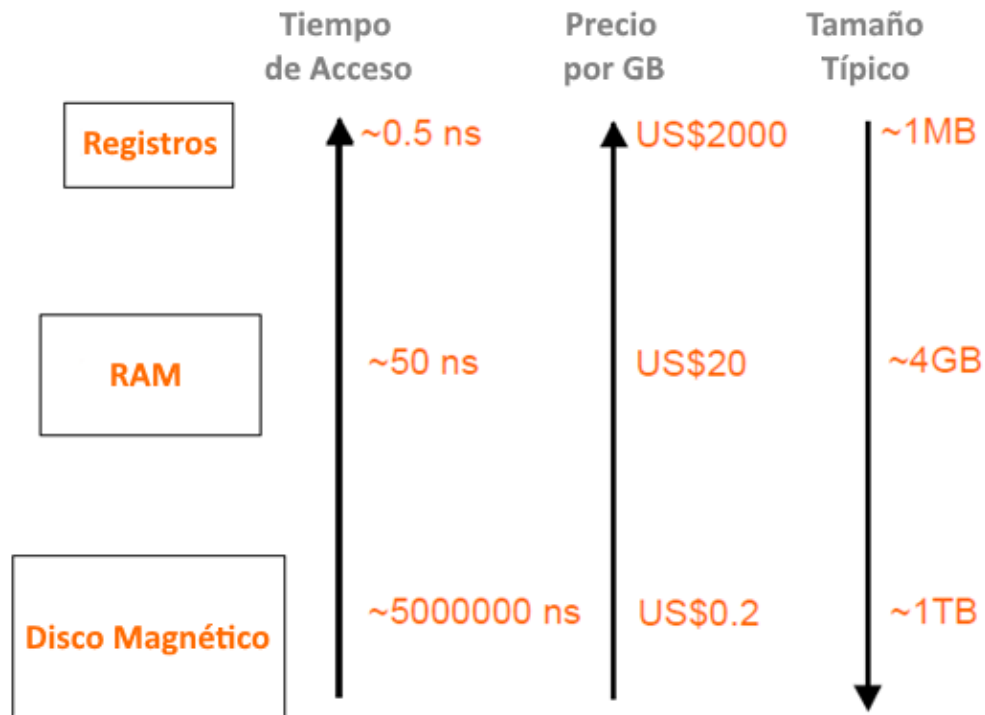




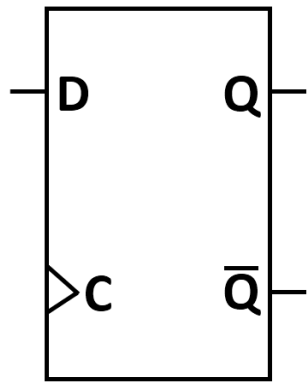




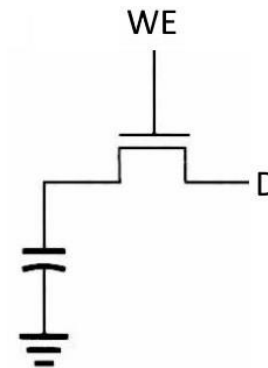
# Revisemos como almacena datos un computador



Almacenamiento **dinámico es más barato**,  
pero requiere refresco de la carga eléctrica



Almacenamiento de 1 bit  
estático (6 transistores)



Almacenamiento de 1 bit  
dinámico(1 transistor)

## Hablemos un poco sobre tipos de dato

- Datos necesitan ser codificados para ser almacenados en memoria.
- Aunque no sea explícito, en un programa cada variable (dato) está asociada a un único tipo de dato en un instante dado.
- El tipo de dato es un atributo de los datos que las características de estos.
- Esto incluye: codificación, tamaños, qué valores pueden tomar, qué operaciones se pueden realizar, etc.

No basta sólo con saber el tipo de dato, se necesita el **endianness**

- Almacenamiento de variables en memoria
  - Endianness: **big endian** o **little endian**

| Address | Value |
|---------|-------|
| 00      | 0x0F  |
| 01      | 0xA7  |
| 10      | 0x1D  |
| 11      | 0x23  |

int 32 bits big endian:      0x 0F A7 1D 23

int 32 bits little endian:      0x 23 1D A7 0F

¿Qué pasa entonces con los arreglos?

# Arreglos son algo más complejos que las variables

- Los arreglos son un grupo ordenado de elementos del mismo tipo
- Para definirlos en memoria se requiere una dirección de inicio, su tipo, el endiannes y el largo (cantidad de elementos)
- Para acceder a los elementos, se requiere un índice que indique su posición
- Indexación depende del tipo y tamaño de cada arreglo

Pontificia Universidad Católica de Chile  
Escuela de Ingeniería  
Departamento de Ciencia de la Computación



# IIC2343 – Arquitectura de Computadores

Almacenamiento de Datos

**Profesor:** Hans Löbel